

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

BLUE SPIKE, LLC,

Plaintiff,

v.

BLU PRODUCTS, INC.

Defendant.

§
§
§
§
§
§
§
§
§
§
§

Civil Action No. 6:16-cv-271

JURY TRIAL DEMANDED

BLUE SPIKE, LLC'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF CONTENTS.....	I
I. INTRODUCTION	1
A. THE INVENTIONS.....	2
II. APPLICABLE CLAIM CONSTRUCTION STANDARDS.....	3
III. LEVEL OF ORDINARY SKILL IN THE ART	4
IV. ARGUMENT	5
1. <i>"copy protecting a software application"</i>	5
2. <i>"during execution of the software application"; "when executing said software application"</i>	7
3. <i>"intermittently relocating"; "intermittently"; "relocating"</i>	9
4. <i>"a software application"; "application"</i>	11
5. <i>"memory scheduler"; "memory scheduler code resource"</i>	13
6. <i>"call"</i>	13
7. <i>"shuffle"</i>	14
8. <i>"randomize"</i>	15
9. <i>"relocate"</i>	16
10. <i>"a program"</i>	16
V. CONCLUSION	18
CERTIFICATE OF SERVICE.....	19

I. INTRODUCTION

The Patents-in-Suit teach novel ways of protecting computer software. See U.S. Patent 5,745,569 (the '569 Patent), *attached as* Ex. 1; U.S. Patent 8,930,719 (the '719 Patent), *attached as* Ex. 2. At the time of the Patents-in-Suit were filed, “protection of intellectual property [had] become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software.” '569 Patent, Col. 1, ll. 5-9. It was in response to this growing need for increased software protection that the inventors created the claims asserted in this case.

Prior to the Patents-in-Suit, computers were designed with a security vulnerability in which software was placed in a fixed location in a computer system's memory. Over time, it became apparent that an attacker could discern a program's fixed location in memory and use that knowledge to either bypass security measures or otherwise manipulate the system in an unintended way. Even software with embedded countermeasures could be compromised because “a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a ‘patch’.” '569 Patent, Col. 7, ll. 41-44. It became apparent to the inventors of the Patents-in-Suit that as long as software remained in a fixed position, it was vulnerable to this type of attack.

In response to this security threat in which an attacker could manipulate a computer system by learning the contents of its memory, the inventors created a novel solution that thwarted an attacker by relocating software code into random positions within memory. Whether relocation occurred only once or more frequently, it provided a much greater degree of security by neutralizing an attacker's attempt learning about the system and bypass security.

This innovation was adopted by the software community within five or six years of the '569 Patent's filing date, particularly by a security endeavor known as the PaX Linux project. That PaX project labeled its implementation of the Patents-in-Suit as "Address Space Layout Randomization" or "ASLR." Today, ASLR is ubiquitous in modern operating systems. And one of the early contributors and creators of ASLR technology, R. Sekar, has already testified that the Patents-in-Suit indeed read on the ASLR technology.

A. The Inventions.

The patents-in-suit each teach how security may be improved by relocating software in memory. The first of the two patents is U.S. Patent 5,745,569 titled "Method for Stega-Cipher Protection of Computer Code." The '569 Patent was filed January, 17, 1996. The single claim asserted from the '569 Patent is Claim 16, which teaches "[a] method for copy protecting a software application executed by a computer system" in which the software is

intermittently relocated. When software is relocated in memory it may be protected from an attacker attempting to bypass copy protection.

The other patent asserted in this suit is U.S. Patent 8,930,719 titled “Data Protection Method and Device” with a priority date of March 24, 1998. ’719 Patent. The ’569 and ’719 patents are not of the same family, but the ’719 Patent shares much of the ’569 Patent’s specification and incorporates it by reference. ’719 Patent, Col. 1, ll. 6-17. Like the ’569 Patent, the ’719 Patent teaches how a computer’s memory may be reorganized. *See, e.g.*, ’719 Patent, Claim 1 (in which “said memory scheduler, when called, functions to shuffle said other code resources in said memory”). However, whereas the ’569 Patent discloses only method claims, the ’719 Patent discloses system claims. Moreover, the ’719 Patent is not limited to copyright protection, teaches a number of other novel improvements, and so on.

II. APPLICABLE CLAIM CONSTRUCTION STANDARDS

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’” *Lennon Image Techs., LLC v. Macy’s Inc.*, 2014 U.S. Dist. LEXIS 105224, at *6 (E.D. Tex. Aug. 1, 2014); *Light Transformation Techs. LLC v. Lighting Sci. Group Corp.*, 2014 U.S. Dist. LEXIS 94090, at *10 (E.D. Tex. July 10, 2014) (*quoting Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc)). The specification “is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of

a disputed term.” *Light Transformation Techs. LLC*, 2014 U.S. Dist. LEXIS 94090, at *11. The prosecution history also supplies intrinsic evidence if it is in evidence. *Lennon Image Tech., LLC*, 2014 U.S. Dist. LEXIS 105224 at *7.

“Differences among the claim terms can also assist in understanding a term’s meaning For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation.” *Alcatel United States Res., Inc. v. Microsoft Corp.*, 2008 U.S. Dist. LEXIS 49615, at *5 (E.D. Tex. Jun. 27, 2008). “Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent.” *Id.* at *7. Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

III. LEVEL OF ORDINARY SKILL IN THE ART

Blue Spike, LLC (“Blue Spike”) proposes that a person of ordinary skill in the art would have a Master’s degree in computer science or computer engineering, or equivalent experience, as well as two years experience in the field of digital signal processing and cryptography.

IV. ARGUMENT

1. “copy protecting a software application”

Blue Spike’s Construction	Defendants’ Construction
<p><u>Proposed Construction:</u> Plain and Ordinary Meaning</p> <p><u>Alternate Construction:</u> “protecting against the unauthorized copying, unauthorized use, or unintended use of a software application.”</p>	<p>“protecting a software application from unauthorized copying or the use of unauthorized copies”</p>

The term “copy protecting a software application” should not be construed. It is a term that is readily understood in the art and “the lay meaning of this term is the same meaning as that which a person having ordinary skill in the art would attribute to the term.” *See Realtime Data, LLC d/b/a IXO v. Packeteer, Inc., et al.*, Case No. 6:08-cv-144-LED-JDL Dkt. No. 371, at * 28 (E.D. Tex. June 22, 2009) (declining to construe the term “data type”), *attached as* Ex. 3. Construing “copy protecting a software application” will needlessly limit and confuse the term.

Defendants’ proposed construction is too limiting. That construction only applies to two specific situations: protecting a software application from (1) unauthorized copying or (2) the use of unauthorized copies. But these two examples are only a subset of what copy protection covers. This is apparent from the specifications, which provide a variety of other examples, including “attempts at memory capture or object code analysis,” “modifying, in an

unintended manner, the functioning of the application,” “attacks at disabling the system,” and “attempts to tamper” with the software. ’569 Patent, 2:21-26; 3:37-40. Defendants’ construction is simply too restrictive to incorporate the patent’s teachings.

In addition to being overly limiting, Defendants’ proposal is also easily misconstrued. Defendants’ construction implies that a copy must be created for copy protection to be relevant. This is certainly not the case. Indeed, copy protection has also been referred to as “content protection” or “copyright protection”; it is not so limited that it only applies to copies of a protected work and not protected original works.

The potential for confusion is real, as evidenced in *Blue Spike v. Huawei Technologies, et al.*, Case No. 6:13-cv-679. In that case, the Court offered a tentative construction of this term that was more aligned with Blue Spike’s alternative construction here. *See* Tentative Claim Construction, Case No. 6:13-cv-679, *attached as* Ex. 4.¹ But the claim construction proposed by Defendants’ was ultimately adopted in the *Blue Spike v. Huawei* case after the Court believed the parties had reached an agreement during the hearing. *See Blue Spike v. Huawei* Claim Construction Order, Case No. 6:13-cv-679, Dkt. No. 194, at 4 (citing Dkt. No. 174 at 10:22-11:1). However, there was no agreement. Moreover, the prior construction has proven inadequate, as

¹ The *Blue Spike v. Huawei* tentative construction was “protecting software from unauthorized copying or unauthorized use.” *See* Ex. 4. Blue Spike’s current alternative construction is “protecting software from unauthorized copying, unauthorized use, or unintended use,” drawing on the specification’s description of this form of copy protection.

evidenced by Huawei's incorrect assumption that a software copy must be made for copy protection to be relevant.

If the Court decides that this term must be construed, Blue Spike proposes "protecting software from unauthorized copying, unauthorized use, or unintended use." Blue Spike's definition is more true to the plain and ordinary meaning, as defined in the specification and understood by one of ordinary skill in the art. Unlike Defendants' proposal, Blue Spike's alternative proposal does not imply that copy protection is only relevant if a copy has been made. Additionally, Blue Spike's alternative proposal is closer to the *Blue Spike v. Huawei* court's tentative construction. Finally, Blue Spike's proposal incorporates how copy protection may prevent the unintended use of software, as described in the specifications. *See* '569 Patent, Col. 2, ll. 24-26.

2. *"during execution of the software application"; "when executing said software application"*

Blue Spike's Construction	Defendants' Construction
<p><u>Proposed Construction:</u> Plain and ordinary meaning</p> <p><u>Alternate Construction:</u> "at any the time between when a program begins and when it ends"</p>	<p>"while the software application is running"</p>

The plain and ordinary meaning of "executed / execution" and "during execution of the software application" is clear and not in need of clarification. *See* MICROSOFT COMPUTER DICTIONARY, Fifth Edition, 2002 ("In programming, execution implies loading machine code of the program into memory and then

performing the instructions.”), *attached as* Ex. 5. “The words of a claim generally receive their ordinary and customary meaning as understood by one of ordinary skill in the art.” *Phillips*, 415 F.3d at 1312–13 (holding that the lower court “incorrectly limited” a term by construing it as something other than its “ordinary and customary meaning”) (internal quotes omitted).

Therefore, this term should not be construed.

Defendant’s proposal is not consistent with the term’s plain meaning or the intrinsic record. Defendant’s definition may be misconstrued as distinguishing “running” from “loading.” Indeed, running and loading both occur during execution. By way of example, an application is executed from a desktop when a user double clicks on an application. The double click begins execution by loading the program and then running it. During execution of the application, the program will load; the program may stop (i.e. go idle) for a period of time; the program may pass control to a different application, and so on. All of these occur during execution. If the Court adopts the Defendants’ proposed construction, a fact-finder may mistakenly infer that the claims only relate to a specific sub-set of executing software: software that has already loaded, is not idle, remains in control, and so on. None of these limitations are described in the specification or assumed by one of ordinary skill in the art. Thus, the Defendants’ proposal must be denied. *See Motion Games, LLC v. Nintendo Co. Ltd., et al.*, Case No. 6:12-cv-878, at*10 (E.D. Tex. Jan. 16, 2015)

(declining to construe the term “object” with limiting language “in view of the absence of any evidence which supports limiting ‘object’”).

3. *“intermittently relocating”; “intermittently”; “relocating”*

Blue Spike’s Construction	Defendants’ Construction
<p><u>Proposed Construction</u> Plain and ordinary meaning</p> <p><u>Alternate Construction #1:</u> “relocating one or more times”</p> <p><u>Alternate Construction #2:</u> “moving to a new location one or more times”</p>	<p>“intentionally shuffling at periodic, random or pseudo-random intervals”</p>

The term “intermittently relocating” is readily understood by one of ordinary skill in the art and should not be construed. Defendants’ construction is a prime example of imposing unnecessary limits on otherwise clear and unambiguous terms. For example, Defendants’ proposal replaces a plainly understood phrase with hyper-technical language that may likely be misunderstood. Defendants’ proposal is rendered from a portion of the specification that states how a memory scheduler “can be called periodically, or at random or pseudo random intervals.” ’569 Patent, Col. 8, ll. 4-5. The first problem using this phrase is that “random” and “pseudo random” are virtually identical—“pseudo random” means that something only appears random.² The second problem with the Defendants’ proposal is that it loses a significant

² Something that is pseudo random has the appearance of being random but it technically not. See <https://en.oxforddictionaries.com/definition/pseudorandom> (defining “pseudorandom” as “satisfying one or more statistical tests for randomness but produced by a definite mathematical procedure”).

aspect of the quoted text. The quote is not merely a list of three potential methods of calling the memory scheduler (i.e. periodically, at random, or at pseudo random intervals), but rather a distinction between non-random intervals (i.e. periodically) and random intervals (i.e. random or pseudorandom). Ultimately, the quoted text is telling one of ordinary skill in the art that the memory scheduler may be called randomly or not randomly. And this rendering of the specification is consistent with the definition of “intermittent” at play here. *See* Microsoft Dictionary, 5th Edition, Ex. 5 (defining “intermittent” as “pertaining to something, such as a signal or connection, that is not unbroken, **but occurs at periodic or occasional intervals.**”) Defendants’ construction does not assist a fact-finder in understanding these nuances and will confuse rather than clarify this term.

It is also problematic that Defendants have imported the term “intentionally” into their definition. This term is pulled from the same portion of the specification as listed above: “[A] ‘memory scheduler[]’ can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources.” ’569 Patent, Col. 8, ll. 4-6. It is apparent from this citation that the word “intentional” merely indicates that a memory scheduler shuffles code in memory when it is called. Unfortunately, this is not clear in Defendants’ construction which remains problematic for a number of reasons. First, “intentionally” is not doing any work. Second, the term “intentionally” in the specification refers here to a memory scheduler that

is not referenced in the asserted '569 Claim 16 cited in this case; it is only referenced in some of the '569's dependent claims.³ Reading “intentionally” into this term imports meaning into all claims rather than the claims that specifically mention a memory scheduler. Third, the word “intentionally” risks confusing the jury as to whom or what is intending the shuffling to occur. A fact-finder may mistakenly intuit that a computer user must intend for the shuffling to occur—an absurd result. Thus, this term should not be construed.

If the Court decides to construe this term, Blue Spike proposes the construction “relocating one or more times.” This definition is more understandable and does more work than “can be called periodically, or at random or pseudo random intervals.” Also, because intervals may occur one or more times, this definition prevents further confusion. *See* Case No. 6:13-cv-679, Dkt. No. 194, at 10 (with regards to the '569 patent, holding that “[t]he patentee’s statement could include multiple shuffles but does not necessarily exclude a single shuffle”).

Alternatively, the Court may construe the term as “moving to a new location one or more times.” Here, the unambiguous term “shuffle” is defined synonymously as “moving to a new location.”

4. “a software application”; “application”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction</u>	“a software program run by an operating system”

³ The term “intermittently relocating” is not used in the '719 Patent claims.

<p>“a computer program”</p> <p><u>Alternate Construction:</u></p> <p>Plain and Ordinary meaning.</p>	
--	--

A claim term is afforded its plain and ordinary meaning unless (1) a patentee disavows the full scope of a claim or (2) “when a patentee sets out a definition and acts as his own lexicographer.” *Blitzsafe Texas, LLC v. Honda Motor Co., Ltd., et al.*, Case No. 2:15-cv-1274, at *8 (E.D. Tex. Sept. 12, 2016). “To act as his own lexicographer, the patentee must ‘clearly set forth a definition of the claim term’, and ‘clearly express an intent to define the term’.” This is exactly what the inventors have done in these patents. The inventors explain that “[a]n executable computer program is variously referred to as an application.” ’569 Patent, Col. 3, ll. 44-45. Here, the inventors clearly define an “application” as an executable “computer program.” Their intent to define “application” should be respected.

If the Court decides not to accept Plaintiff’s definition as defined by the inventors, the term should not be construed. A “software application” is “familiar to users of computers.” ’569 Patent, 5:27-31. Defendants’ proposed construction again unnecessarily limits the term’s plain and ordinary meaning. There is no indication the inventors wanted to distinguish an operating system from a software application, as both are executable computer programs from a user’s point of view. *See* IBM SYSTEMS REFERENCE LIBRARY, Fifth Edition, 1972 (“An operating system is an application of a computing

system.”), *attached as* Ex. 2. It was the inventors’ intention to construe this term broadly, and it should not be redefined. *See Thorner v. Sony Computer Entm’t Am. LLC*, 669 F.3d 1362, at 1366-67 (Fed. Cir. 2012) (“[W]e do not redefine words. Only the patentee can do that.”).

5. “*memory scheduler*”; “memory scheduler code resource”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u> Plain and ordinary meaning.	<u>Proposed Construction:</u> Means-plus-function limitation.

The term “memory scheduler” requires no construction. Defendants argue “memory scheduler” is a means-plus-function limitation. This is incorrect. A scheduler is “[a] computer program designed to perform functions such as scheduling, initiation, and termination of jobs.” Here, “memory” in “memory scheduler” identifies the scheduler the invention is referring to: a scheduler of RAM or a “memory scheduler.” This would be readily understood by one of ordinary skill in the art.

6. “*call*”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u> Plain and ordinary meaning. <u>Alternate Construction:</u> “A statement in a computer program	<u>Proposed Construction:</u> “Transfer of control during execution time.”

that references a subroutine or program.”	
---	--

The term “call” is another term that requires no construction. One of ordinary skill in the art would understand that when a program is called it is triggered to perform its assigned function. Defendants’ suggestion does not clarify the term, and it also introduces confusion by implying that the underlying program is no longer in control, has ended, and so on. Additionally, Defendants’ insistence on referring to “execution time” in this phrase and “during program run time” in other phrases (e.g. shuffle, randomize, and relocate) only causes additional confusion.

If the Court decides to construe this term, Blue Spike proposes “call” be construed as “a statement in a computer program that references a subroutine or program.” Blue Spike’s construction resolves the ambiguity inherent in Defendants’ construction.

7. “shuffle”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u> Plain and ordinary meaning. <u>Alternate Construction:</u> “Move a portion of a sequence to a new location in a sequence.”	<u>Proposed Construction:</u> “Randomly reorganize during program run time.”

“Shuffle” needs no construction. And like Defendants’ other proposed constructions, its proposed construction for “shuffle” is problematic for a variety of reasons. First, Defendants’ construction is more confusing than the original term. Second, Defendants’ construction improperly imports the idea of “during program run time.” As explained above, Defendants’ citation to “program run time” is misleading and unnecessarily limiting. Third, the term “shuffle” is only used in the claims of the ’719 Patent where “randomize” is also used. Defendants’ attempt to construe “shuffle,” “randomize,” and “relocate” the same undermines the inventors’ desire to differentiate their claims. If the inventors had wanted shuffling to be random, they would have said so.

If the Court decides to construe this term, Blue Spike proposes the construction “move a portion of a sequence to a new location in a sequence.” This construction embodies the idea that a collection of code resources may be moved to different locations and affords “shuffle” its own meaning rather than sharing it with disparate terms.

8. “randomize”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u> “employ random selection”	<u>Proposed Construction:</u> “Randomly reorganize during program run time.”

The plain and ordinary meaning of “randomize” is to “employ random selection.” Of the two proposed constructions, only Blue Spike’s proposal is a

viable solution. As mentioned above, “shuffle” and “randomize” are both cited in the ’719 Patent. Therefore, they should be given distinct meanings. Here, Defendant is again attempting to employ the same definition for “randomize” as for “shuffle” and “relocate.” Also as noted above, the two terms are not used synonymously. Finally, this definition again improperly imports additional meaning with the phrase “during program runtime.” There is no indication that the inventors wanted this to be a part of the term, and it is not what one of ordinary skill in the art would necessarily require.

9. “relocate”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u> “move to a new location”	<u>Proposed Construction:</u> “Randomly reorganize during program run time.

The plain and ordinary meaning of “relocate” is to “move to a new location.” This is the third of three distinct terms that Defendants’ would like to construe the same, ignoring the inventors’ intention of differentiating their claims. This cannot be allowed. The term “relocate” does not require randomness, nor must it occur during run time.

10. “a program”

Blue Spike’s Construction	Defendants’ Construction
<u>Proposed Construction:</u>	A set of instructions run by an operating system.

Plain and ordinary meaning.	
-----------------------------	--

The term “a program” does not need to be construed. Defendants’ proposal is another attempt to distinguish the operating system as something separate from a program. But Defendants’ proposal is not only contrary to the understanding of one of ordinary skill in the art, it is illogical. An operating system is a program, developed by software developers just as a program is developed. Granted, an operating system is a specialized program that has the unique task of interacting with the computer’s devices, memory, etc., but it is created like, modified like, and executed like any other program. One of ordinary skill in the art would consider an operating system a program. Defendants’ construction improperly attempts to make a distinction between a program and operating system that effectively excludes operating systems from the claim language. This should not be allowed.

V. CONCLUSION

Blue Spike's proposed constructions allow the detailed intrinsic record to speak for itself. Conversely, Defendants' proposed constructions are unnecessarily limiting and obstruct rather than clarify the terms' meaning. For these reasons, Blue Spike respectfully asks the Court to adopt its constructions.

Respectfully submitted,

/s/ Randall T. Garteiser

Randall T. Garteiser

Lead Attorney

Texas Bar No. 24038912

rgarteiser@ghiplaw.com

Christopher A. Honea

Texas Bar No. 24059967

chonea@ghiplaw.com

GARTEISER HONEA, P.C.

218 North College Avenue

Tyler, Texas 75702

(903) 705-0828

(903) 526-5477 fax

Kirk J. Anderson

California Bar No. 289043

kanderson@ghiplaw.com

Ian N. Ramage

California Bar No. 224881

iramage@ghiplaw.com

GARTEISER HONEA, P.C.

44 North San Pedro Road

San Rafael, California 94903

(415) 785-3762

(415) 785-3805 fax

Counsel for Blue Spike, LLC

Certificate of Service

The undersigned certifies that the foregoing document was filed electronically in compliance with Local Rule CV-5(a). As such, this document was served on all counsel who are deemed to have consented to electronic service. Local Rule CV-5(a)(3)(A). Pursuant to Federal Rule of Civil Procedure 5(d) and Local Rule CV-5(d) and (e), all other counsel of record not deemed to have consented to electronic service were served with a true and correct copy of the foregoing by email.

/s/ Randall Garteiser